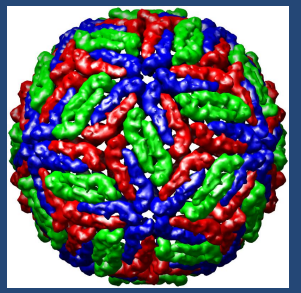


CAPSID

Computational Algorithms for Protein Structures and Interactions

David Ritchie + Isaure Chauvot de Beauchêne
Inria Nancy – Grand Est

Structural Bioinformatics Tools and Techniques

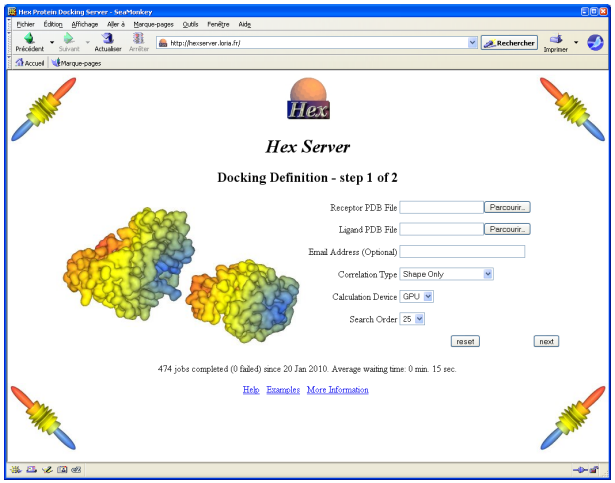
In-House Software

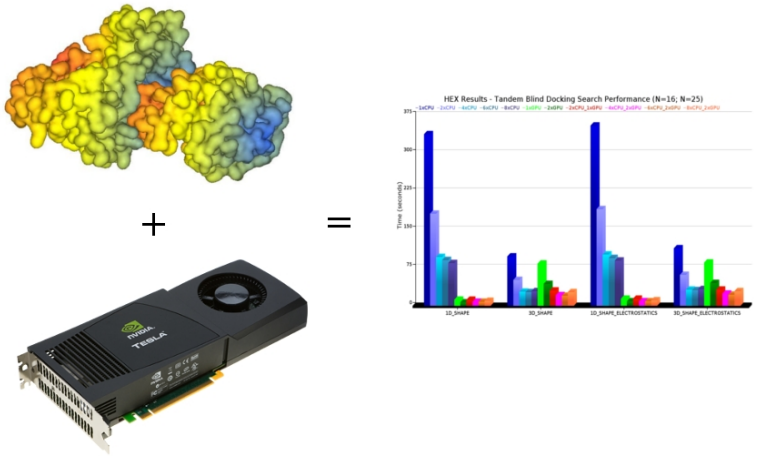
- **Hex**– protein docking by spherical polar FFT
- **Sam** – spherical polar FFT docking of symmetrical complexes
- **gEMfitter** – cryo-EM protein density fitting by FFT on GPU
- **KBDOCK** – database of 3D domain-domain interactions
- **Kpax** – multiple flexible protein structure alignment

External Tools

- Molecular dynamics simulation & modeling: NAMD, Modeller, ...

“Hex” – Spherical Polar Fourier Protein Docking





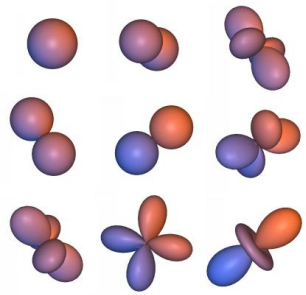
- SPF approach => analytic translational + rotational correlations
- Shape-based scoring function (surface skin overlap volume)
- Can cover 6D search space using 1D, 3D, or 5D rotational FFTs...
- “Easy” to accelerate the 1D FFTs on highly parallel GPUs ...

Sam/Hex: Spherical Polar Fourier Basis Functions

- Represent protein shape as a 3D shape-density function...

$$\tau(\underline{r}) = \sum_{nlm}^{N} a^{\tau}_{nlm} R_{nl}(r)\, y_{lm}(\theta, \phi)$$

- ...using spherical harmonic, $y_{lm}(\theta, \phi)$, and radial, $R_{nl}(r)$, basis functions



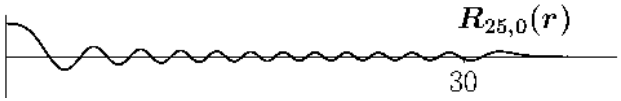
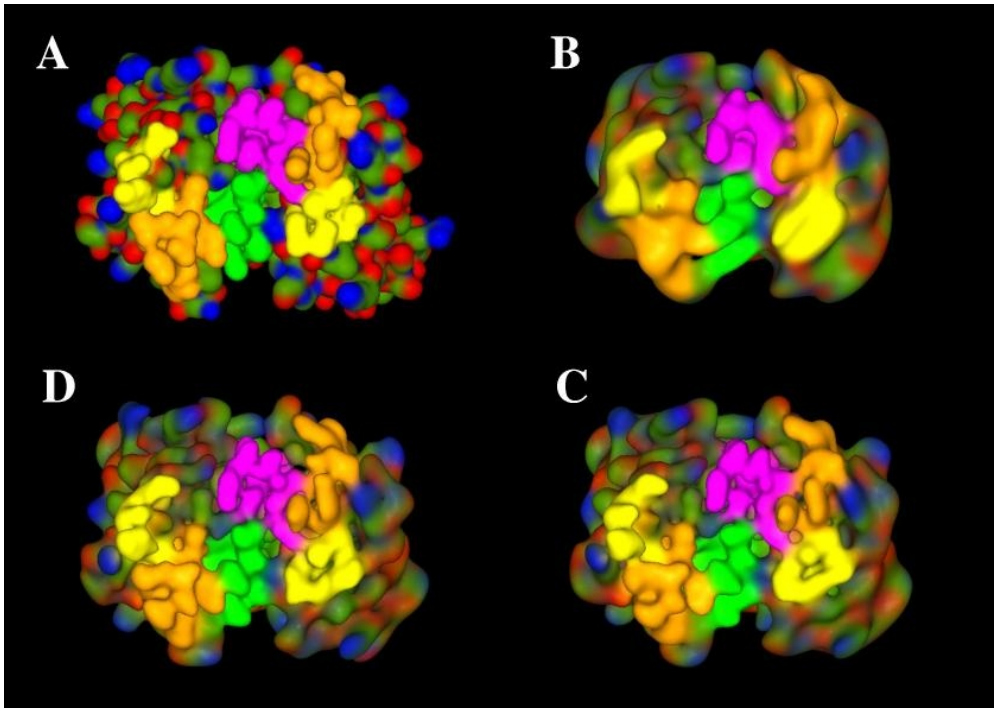
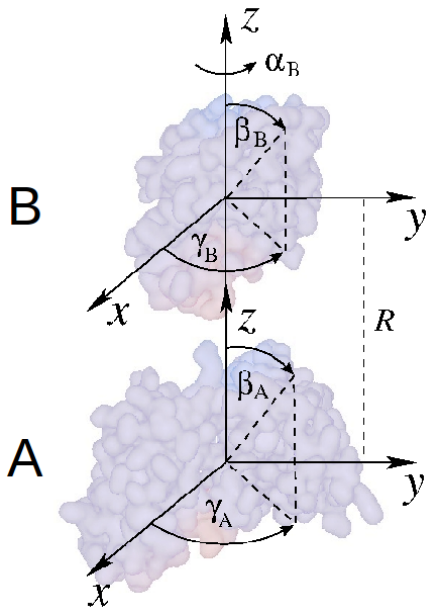


Image	Order	Coefficients
A	Gaussians	-
B	N = 16	1,496
C	N = 25	5,525
D	N = 30	9,455



Coordinate Operators and Docking Equations

- Polar Fourier basis is "natural" for rotational search problems



Describe search space using operators

- Rotation: $\hat{R}(\alpha, \beta, \gamma) = \hat{R}_z(\alpha)\hat{R}_y(\beta)\hat{R}_z(\gamma)$
- Translation: $\hat{T}_z(R)$

Describe interaction as an "equation"

- $\hat{R}(0, \beta_A, \gamma_A)A(\underline{r}) \longleftrightarrow \hat{T}_z(R)\hat{R}(\alpha_B, \beta_B, \gamma_B)B(\underline{r})$

Can re-write this in many ways, e.g.

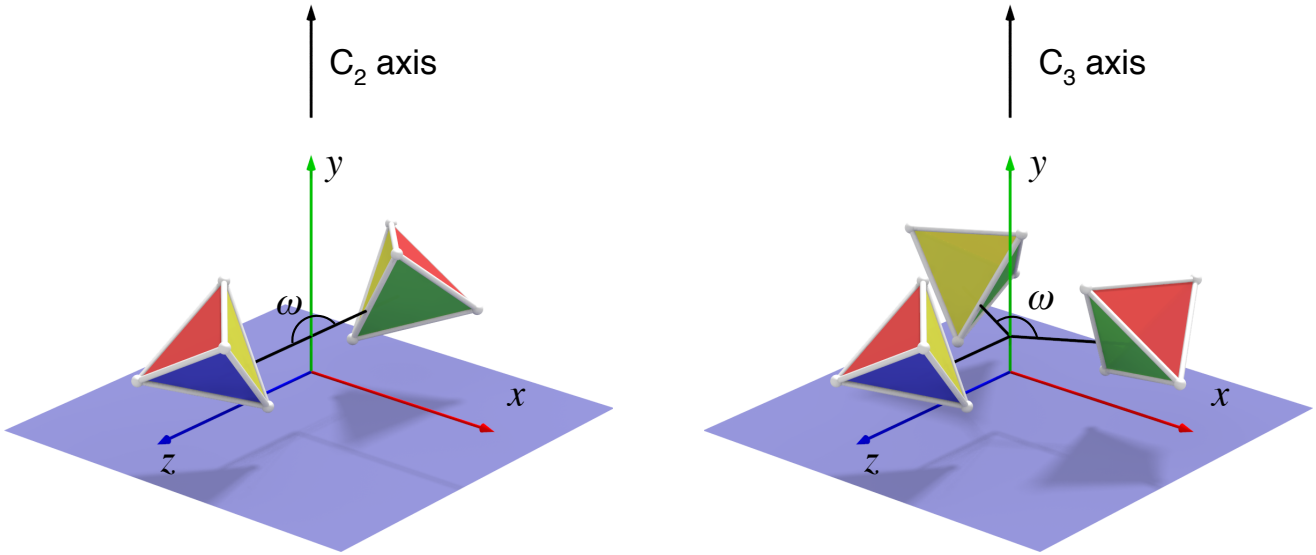
- $\hat{R}(\alpha_B, \beta_B, 0)^{-1}\hat{T}_z(R)^{-1}\hat{R}(0, \beta_A, \gamma_A)A(\underline{r}) \longleftrightarrow \hat{R}_z(\gamma_B)B(\underline{r})$

- Ultimately, operators transform coefficients in "simple" ways, e.g.

Score: $$S_{AB}(\gamma_B) = \sum_{nlmp} \left[A'_{nlm}\right].\left[B^*_{nlp}e^{-ip\gamma_B}\right].$$

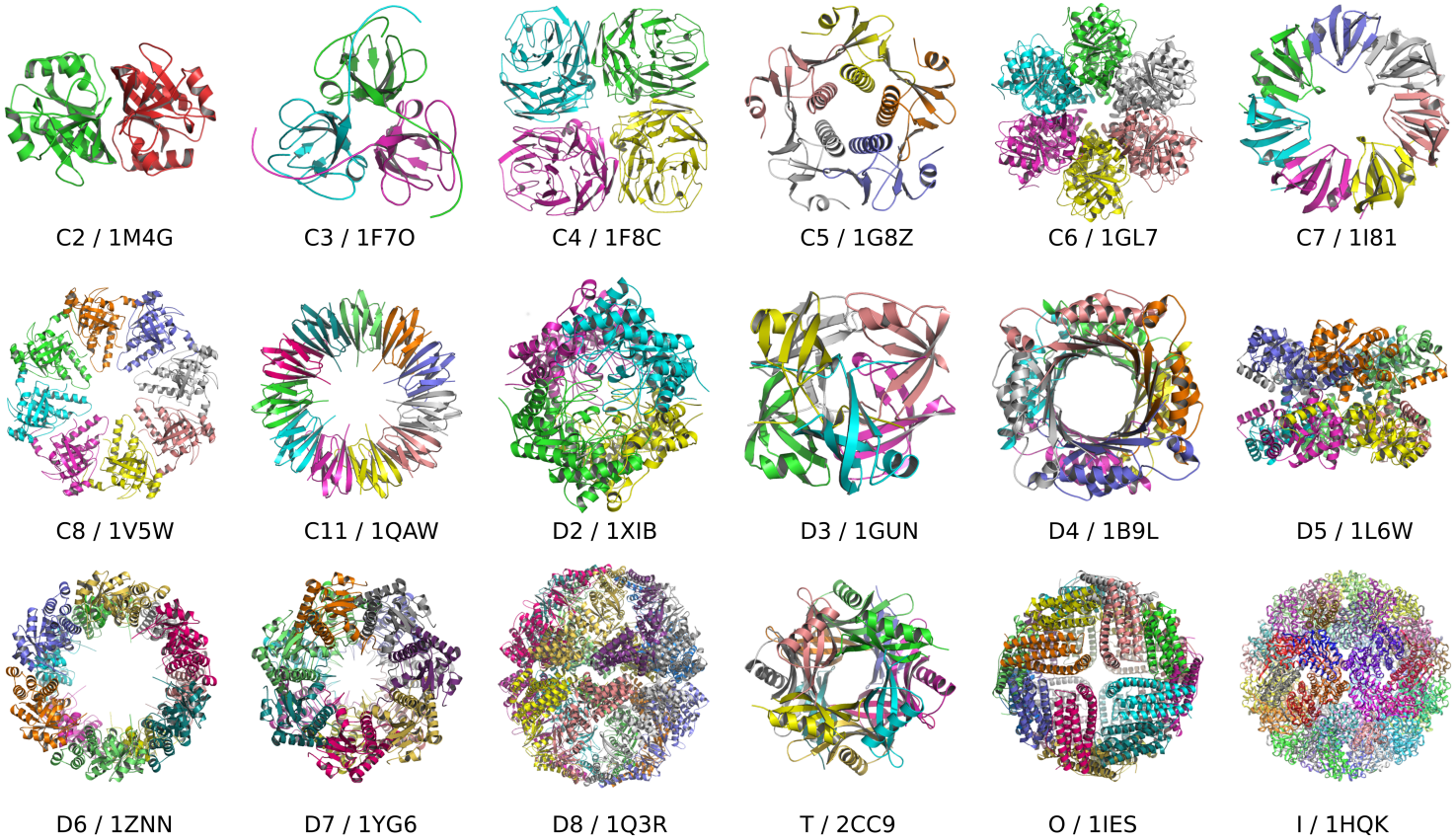
The Docking Equation for Cyclic Symmetries (C_n)

- C_n systems are planar, with symmetry operator $\hat{R}_y(\omega = 2\pi/n)$



- $\hat{R}_y(\omega)\hat{T}_z(D)\hat{R}(\alpha,\beta,\gamma)A(\underline{r}) \longleftrightarrow \hat{T}_z(D)\hat{R}(\alpha,\beta,\gamma)A(\underline{r})$
- After some working, we get a Fourier series in α:
  $S_{AB}(\alpha) = \sum_{nlmp} A_{nlm}(\beta,\gamma)A_{nlp}(D,\beta,\gamma)^* d^{(l)}_{mp}(\omega)e^{-i(p-m)\alpha}$

Sam Results – Examples of Each Symmetry Type

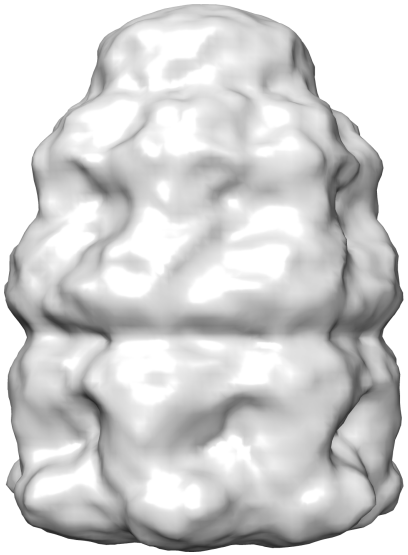


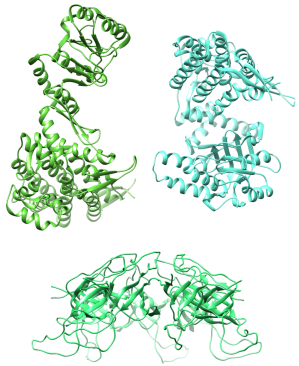
- All except 2 solutions are rank-1, RMSD < 3 Å w.r.t. crystal structure
- Main limitation is size of monomer (approx 500 residue limit)

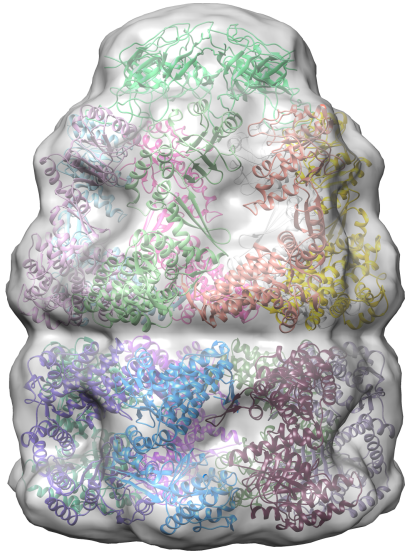
Ritchie and Grudinin (2016), J Appl. Cryst., 49, 158–167

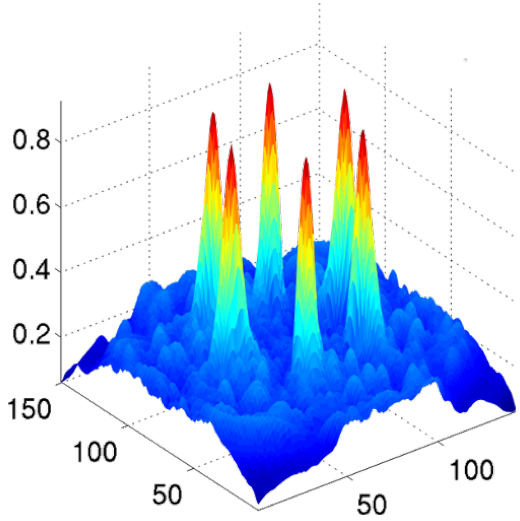
“gEMfitter” – GPU-Accelerated Cryo-EM Density Fitting

- Representation: 3D shape-density in Cartesian grid
- Search: brute force search with FFT acceleration
- Scoring: normalised cross correlation with Laplacian filter





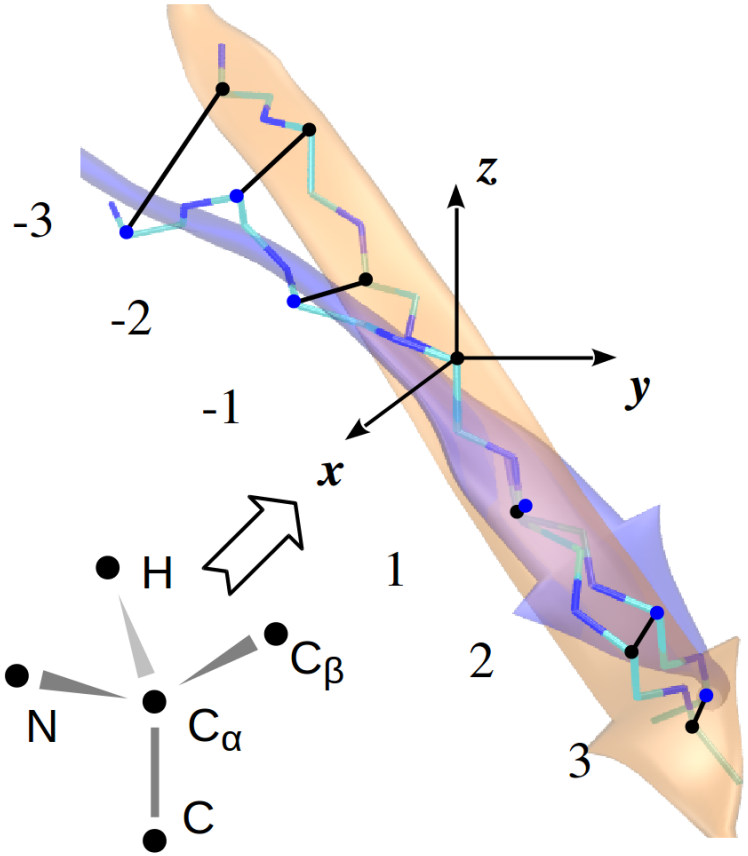




- Calculates 3D translations using Cartesian FFT
- Calculates 3D rotations in GPU texture memory

Kpax – Protein Structure Alignements

- For the first time: exploit the tetrahedral geometry of C_α atoms to superpose pairs of residues *without doing least-squares fitting*



- Score similarity of local environment of residues (i, j) as product of 3D Gaussians between up-stream and down-stream C_α pairs:

$$K_{i,j} = \Pi_{k=-n}^{n} e^{-\beta_k R^2_{i+k,j+k}/4\sigma_k^2}$$

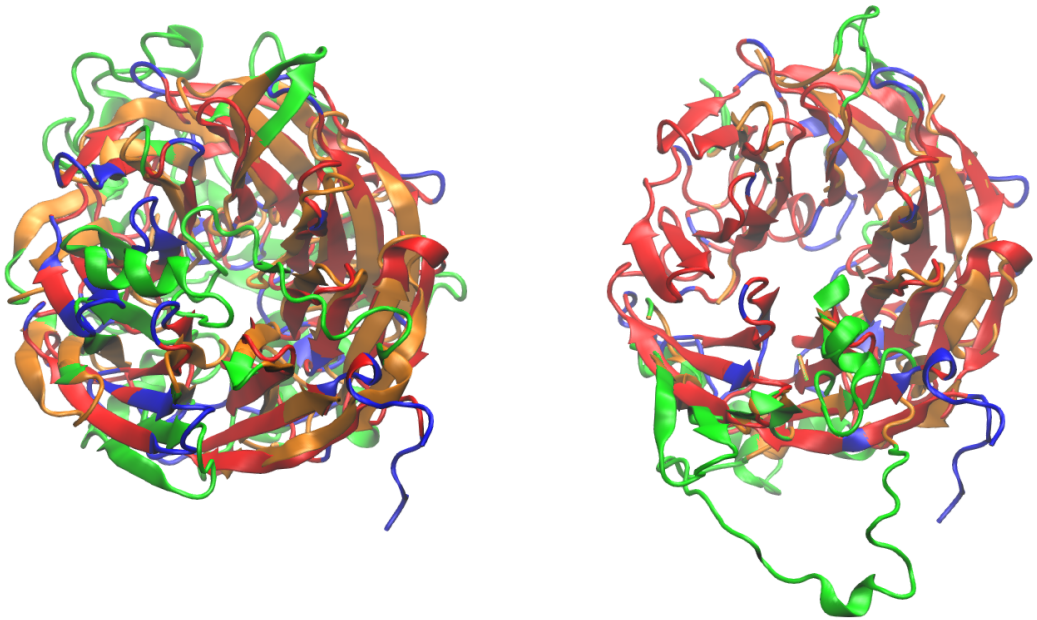
- Gives a very fast way to score local 3D similarity of all residue pairs

Ritchie *et al.* (2012), Bioinformatics, 28, 3274–3281

Results – Comparing Rigid and Flexible Alignments

Example: methyl dehydroxygenase / galactose oxidase

- PDB codes: 4AAH (572 AA; green/orange) and 1GOF (388 AA; blue/red)



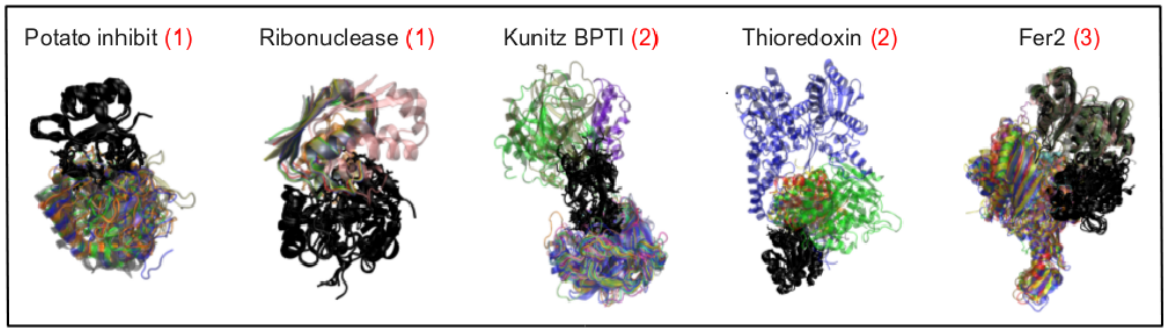
- all red/orange regions are structurally aligned
- left: rigid; 267 pairs, 3.3 Å RMSD (20 identities)
- right: flexible; 308 pairs, 2.2 Å RMSD (23 identities)

- Compare with TM-Align (rigid only):
  - TM-Align: 366 pairs, 5.4 Å RMSD (19 identities)
  - Δ(TM-Align, Kpax): 11.6 Å RMSD

Applications – PDB-Wide Structure Comparison

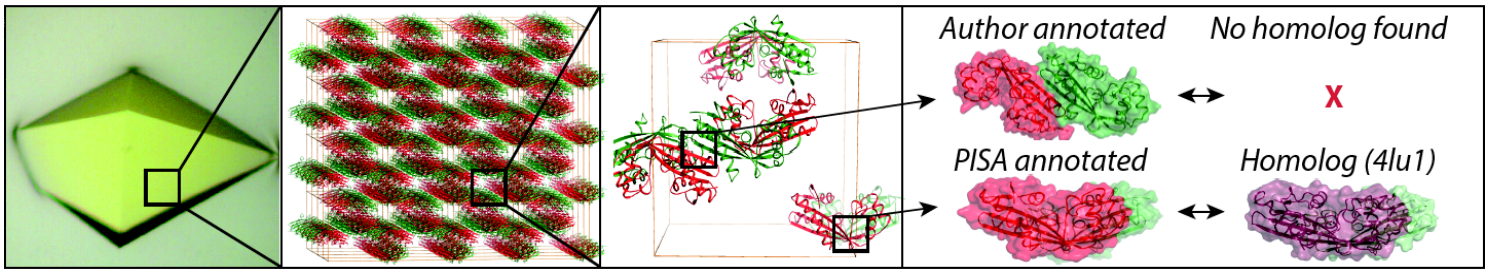
KBDOCK[1] – database of 3D domain-domain interactions

- Allows us to identify "Domain Family Binding Sites" (DFBSs)



QsBio[2] – identifying biologically relevant quaternary structures

- Allows us to predict QS by homology and to fix wrong annotations in PDB



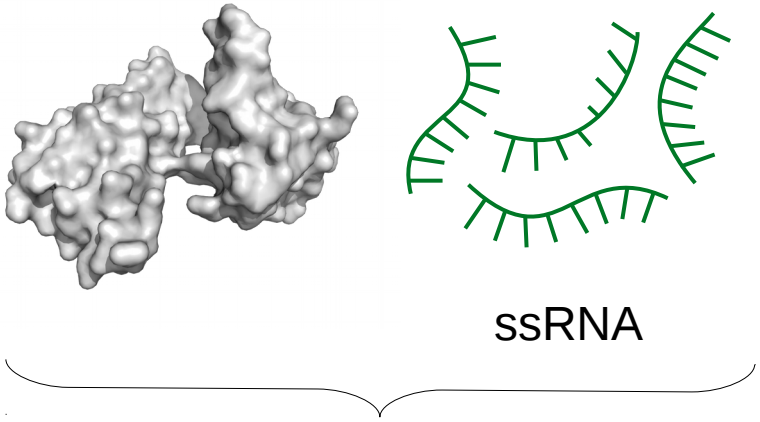
[1] Ghoorah *et al.* (2014), Nucleic Acids Research, 42, D389–D395
[2] Dey, Ritchie, Levy (2017), in press

Thank You!

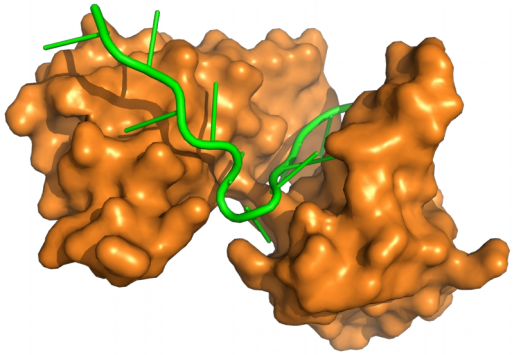
http://capsid.loria.fr/

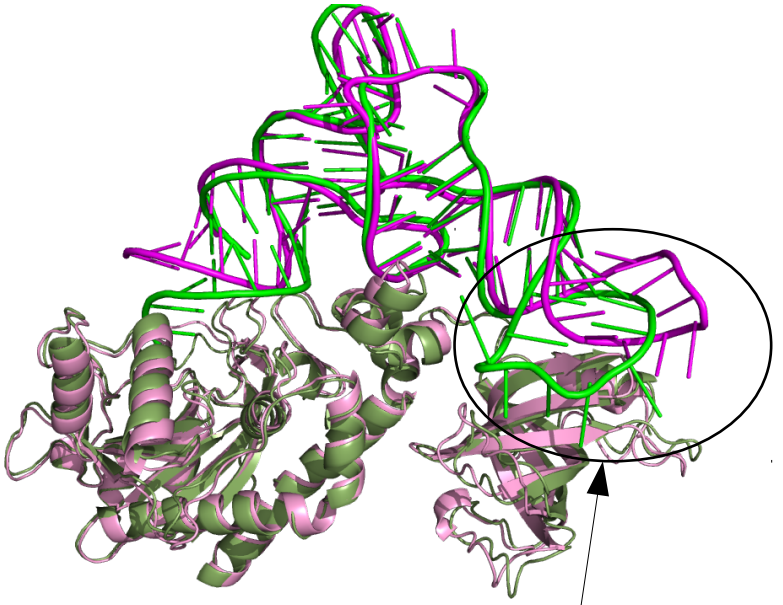
http://hex.loria.fr/
http://sam.loria.fr/
http://gem.loria.fr/
http://kpax.loria.fr/
http://kbdock.loria.fr/

Fragment-based ssRNA docking



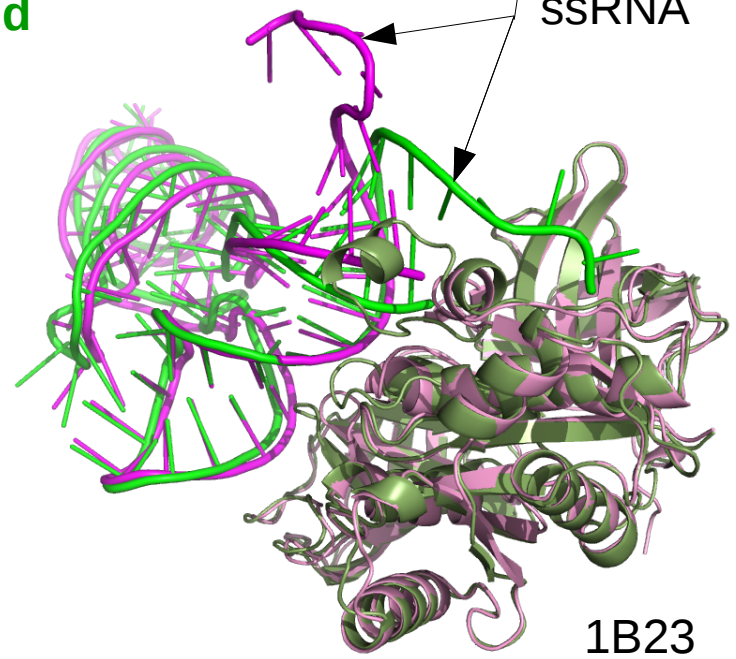
ab initio docking





unbound
vs
bound

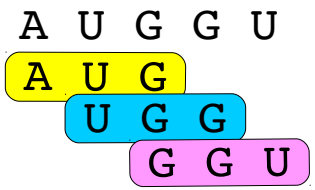
ssRNA



1B23

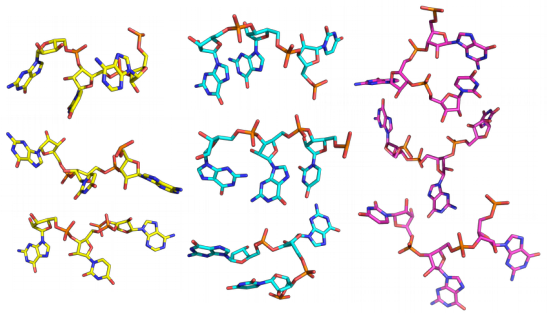
Fragment-based ssRNA docking

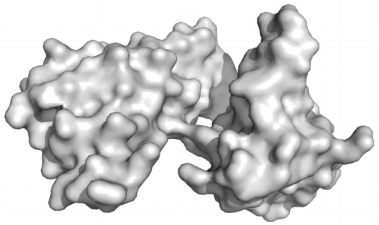
RNA sequence



Fragment library

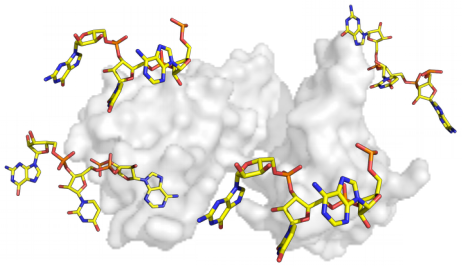
~ 3000 conf per sequence

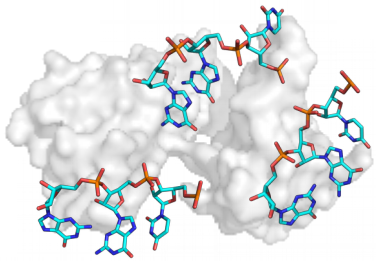


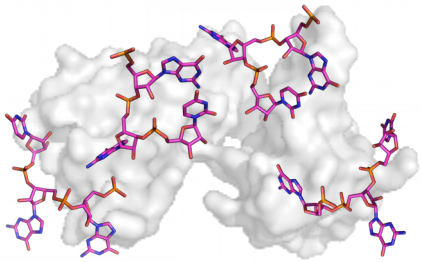


Protein structure

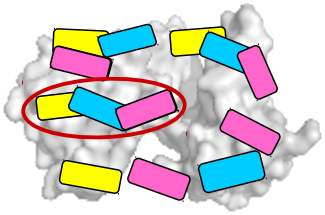
Docking

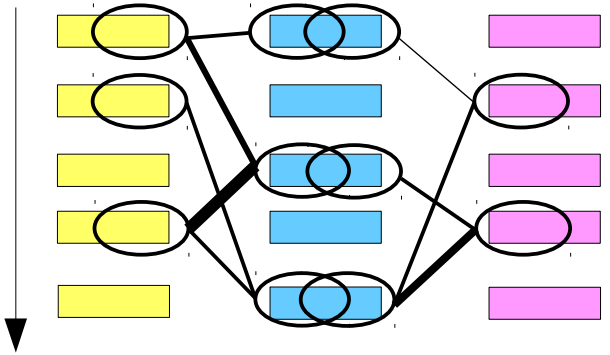






Combinatorial assembly





Energy

~500.000 poses

Search for path with

- Low total energy
- High connectivity
- No clashes

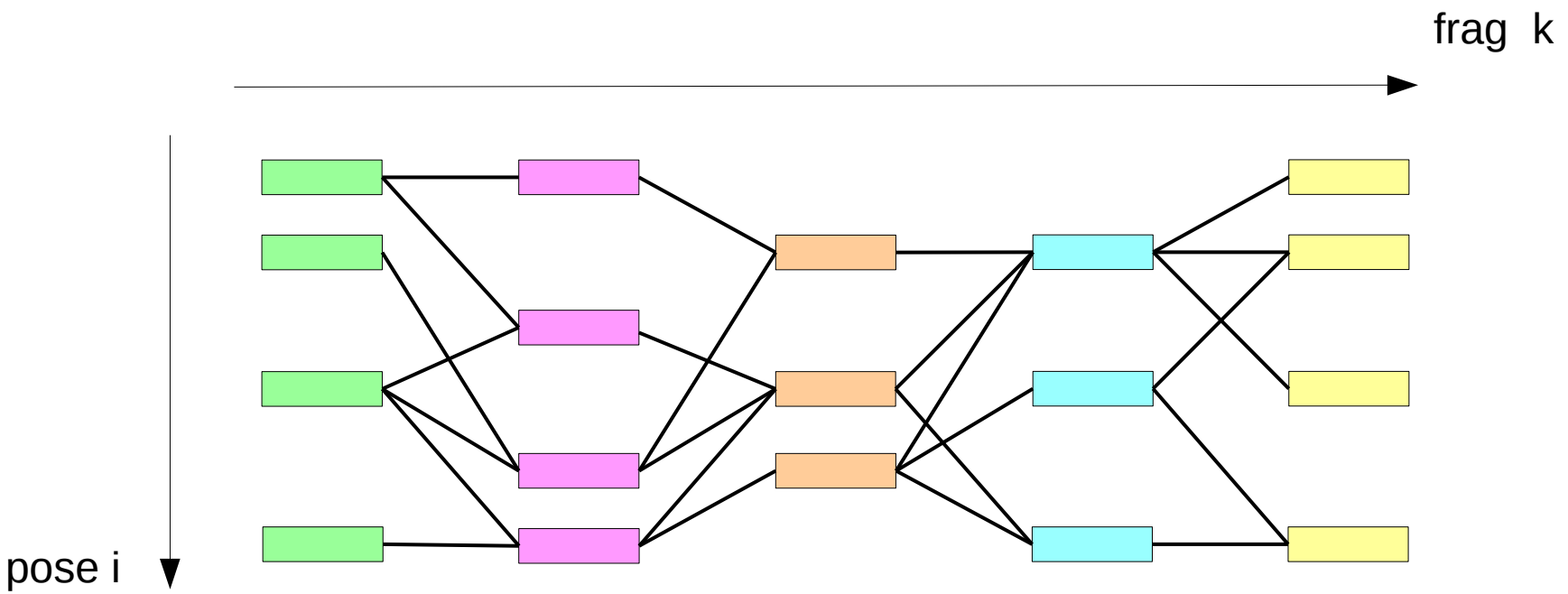
Fragment-based ssRNA docking

Connection propensity

$$N_{fwd}(k,i) = \sum_{i' \in neigbors(i)} N_{fwd}(k+1,i')$$

$$N_{bwd}(k,i) = \sum_{i' tq\, i \in neigbors(i')} N_{bwd}(k-1,i')$$

$$N_{tot}(k,i) = N_{fwd}(k,i) \times N_{bwd}(k,i)$$



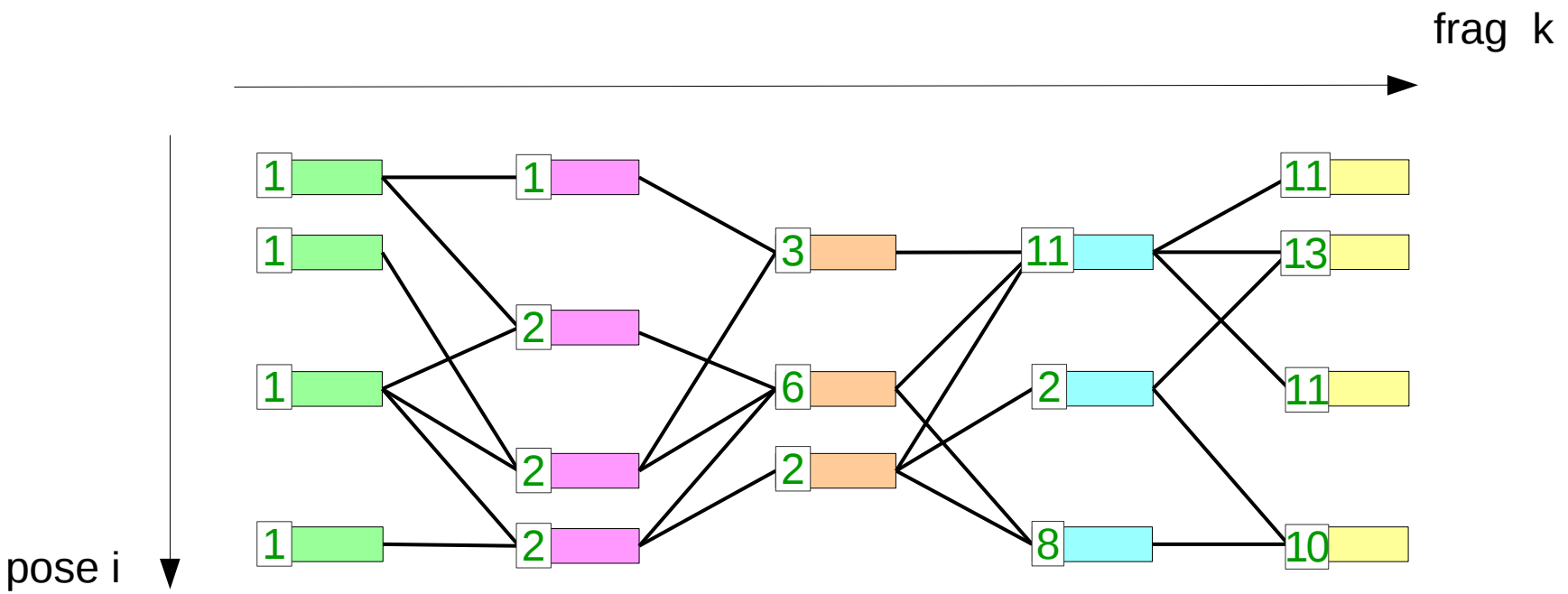
Fragment-based ssRNA docking

Connection propensity

$$N_{fwd}(k,i) = \sum_{i' \in neigbors(i)} N_{fwd}(k+1,i')$$

$$N_{bwd}(k,i) = \sum_{i' tq\, i \in neigbors(i')} N_{bwd}(k-1,i')$$

$$N_{tot}(k,i) = N_{fwd}(k,i) \times N_{bwd}(k,i)$$



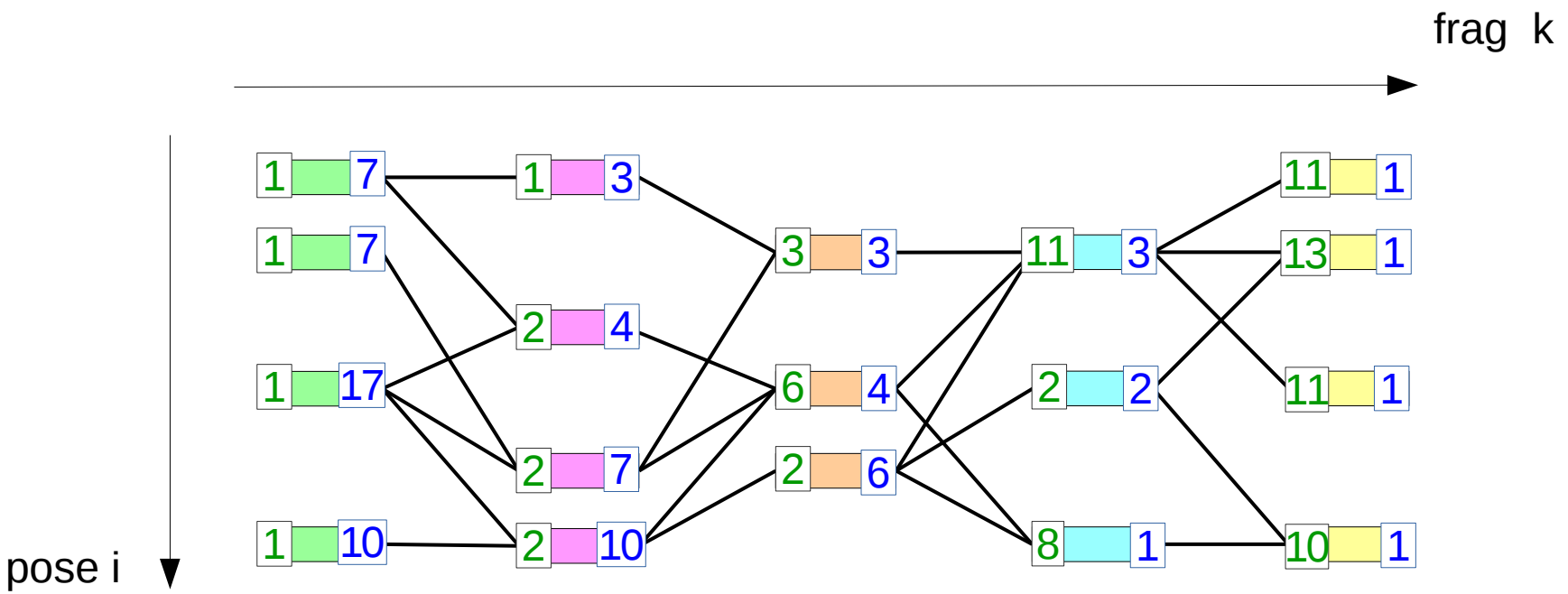
Fragment-based ssRNA docking

Connection propensity

$$N_{fwd}(k,i) = \sum_{i' \in neigbors(i)} N_{fwd}(k+1,i')$$

$$N_{bwd}(k,i) = \sum_{i' tq\, i \in neigbors(i')} N_{bwd}(k-1,i')$$

$$N_{tot}(k,i) = N_{fwd}(k,i) \times N_{bwd}(k,i)$$



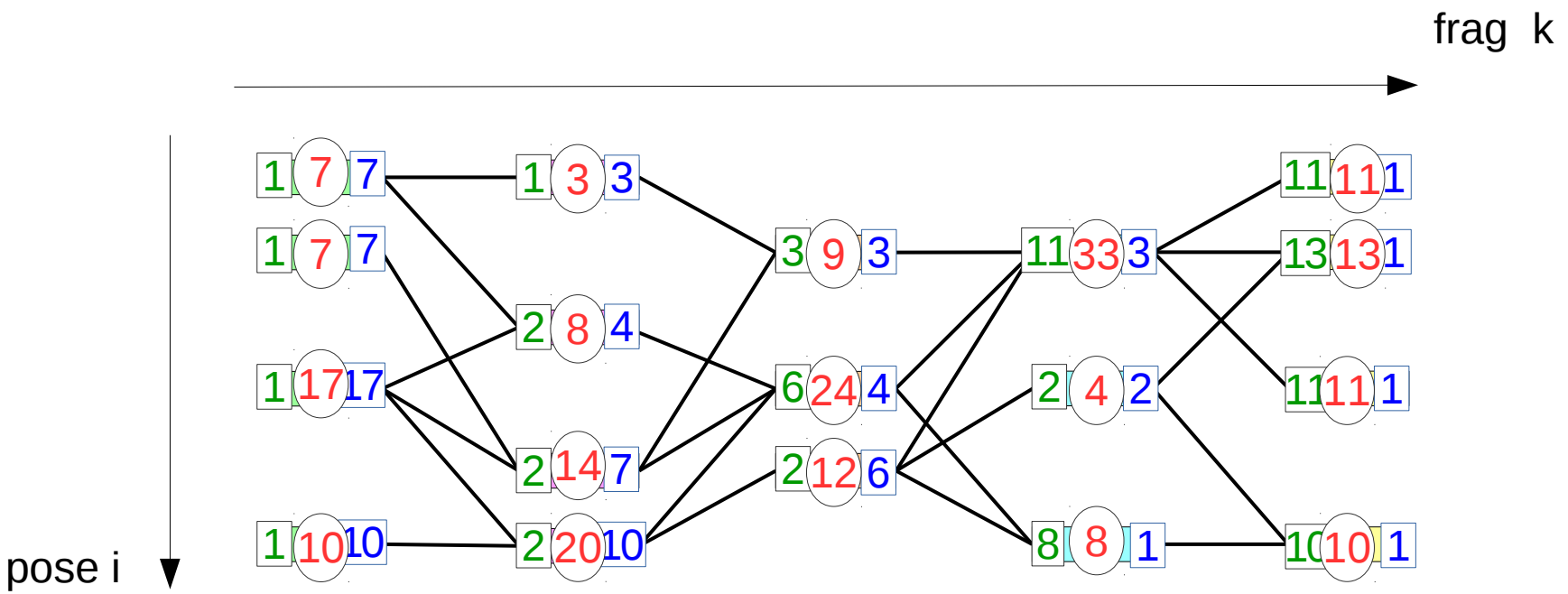
Fragment-based ssRNA docking

Connection propensity

$$N_{fwd}(k,i) = \sum_{i' \in neigbors(i)} N_{fwd}(k+1,i')$$

$$N_{bwd}(k,i) = \sum_{i' tq\, i \in neigbors(i')} N_{bwd}(k-1,i')$$

$$N_{tot}(k,i) = N_{fwd}(k,i) \times N_{bwd}(k,i)$$



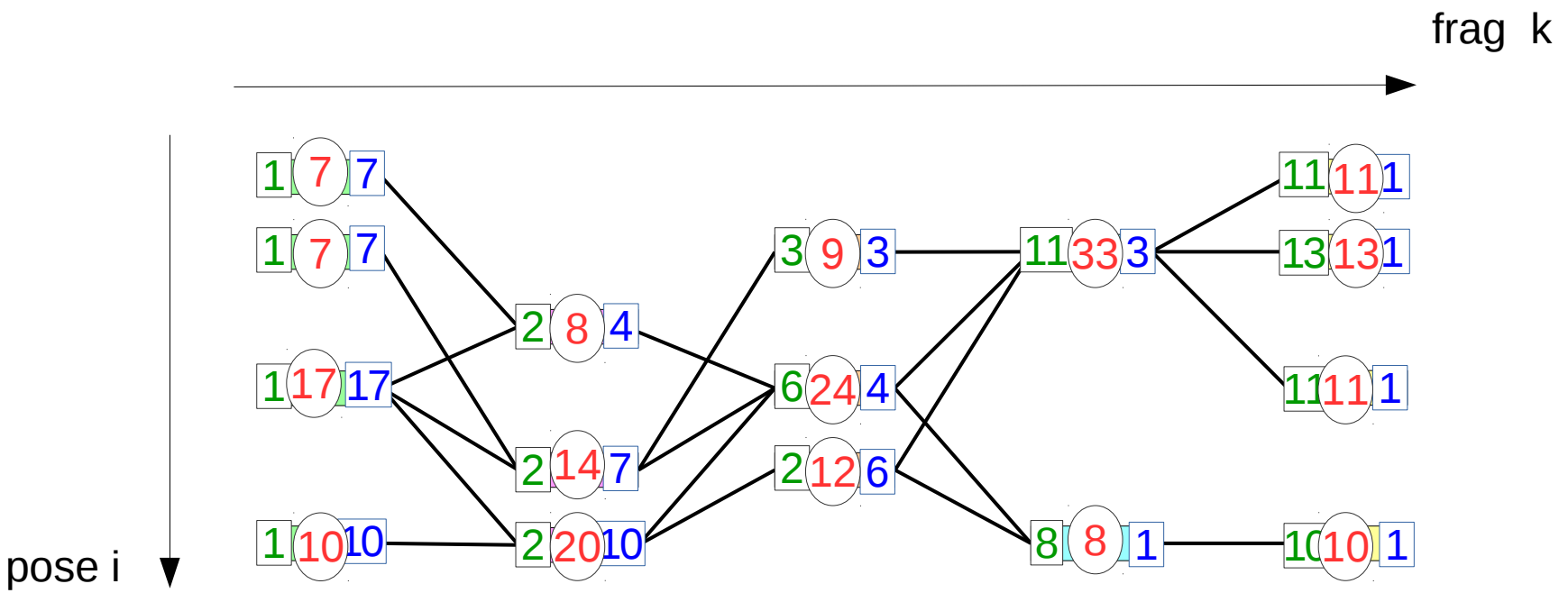
Fragment-based ssRNA docking

Connection propensity

$$N_{fwd}(k,i) = \sum_{i' \in neigbors(i)} N_{fwd}(k+1,i')$$

$$N_{bwd}(k,i) = \sum_{i' tq\, i \in neigbors(i')} N_{bwd}(k-1,i')$$

$$N_{tot}(k,i) = N_{fwd}(k,i) \times N_{bwd}(k,i)$$



Fragment-based ssRNA docking

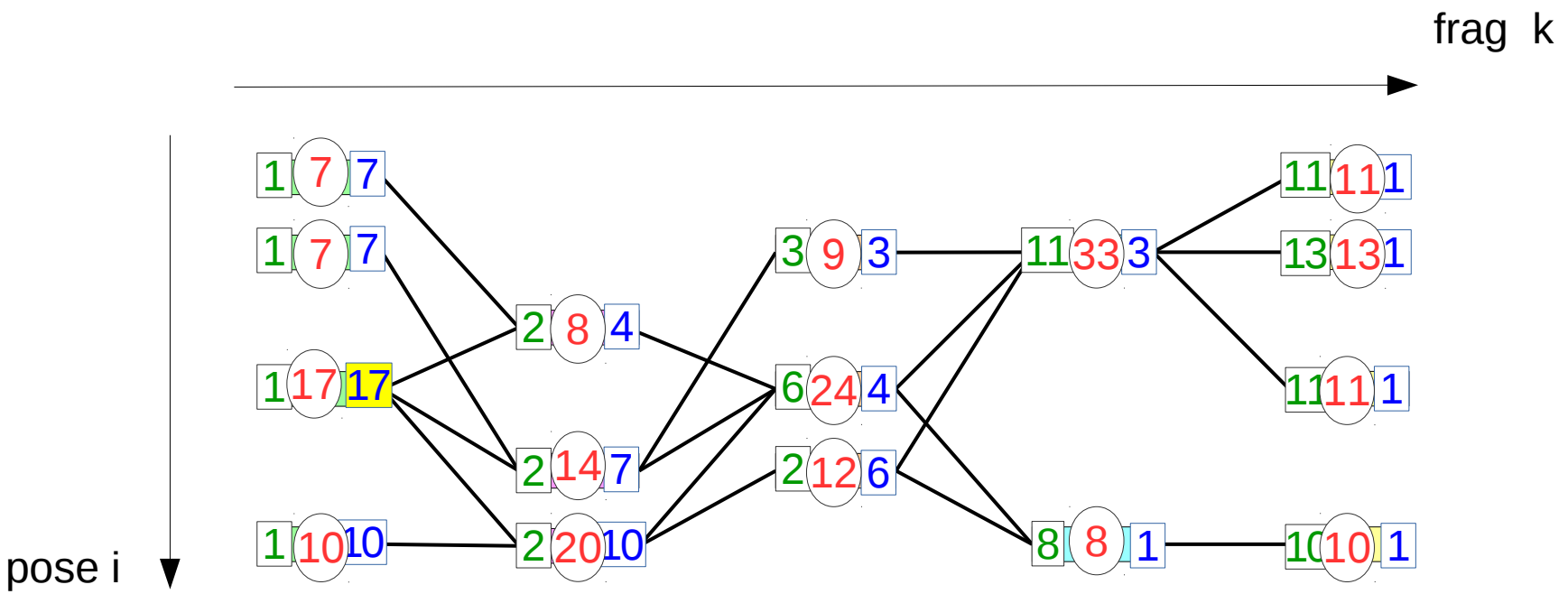
Connection propensity

$$N_{fwd}(k,i) = \sum_{i' \in neigbors(i)} N_{fwd}(k+1,i')$$

$$N_{bwd}(k,i) = \sum_{i' tq\, i \in neigbors(i')} N_{bwd}(k-1,i')$$

$$N_{tot}(k,i) = N_{fwd}(k,i) \times N_{bwd}(k,i)$$

Stochastic backtracking => enumerate chains



Fragment-based ssRNA docking

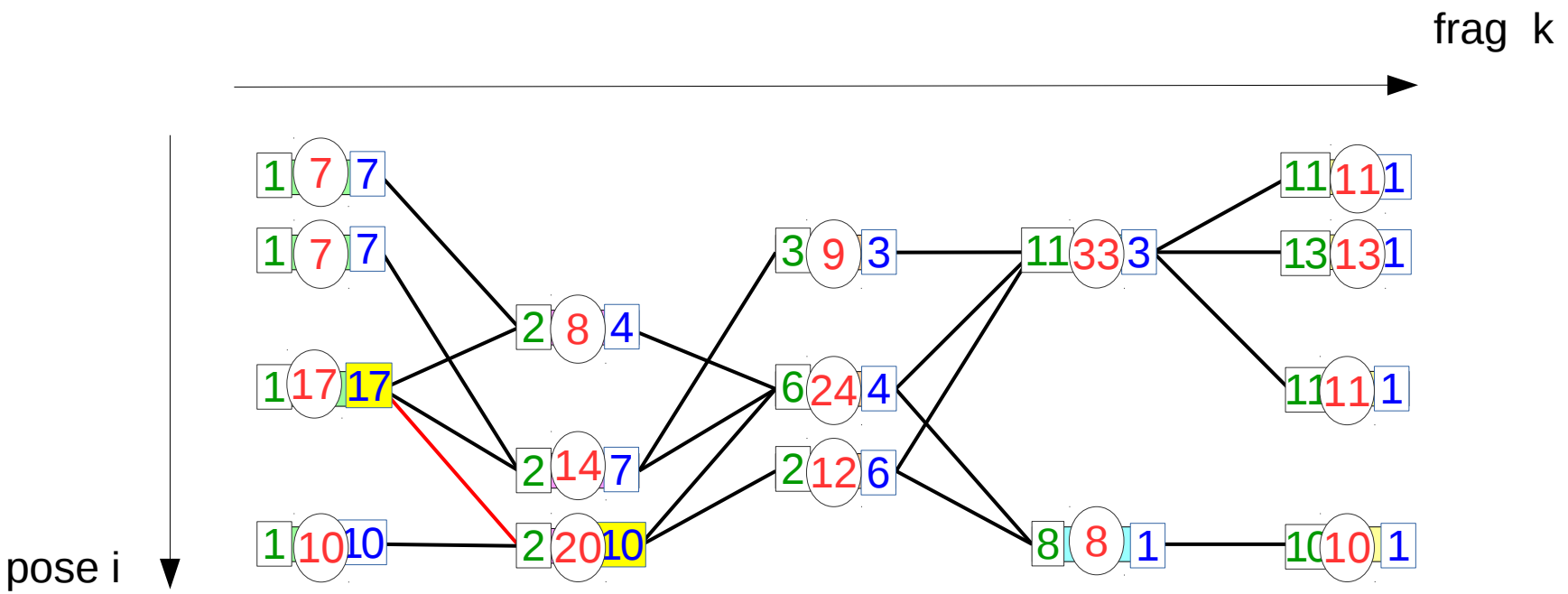
Connection propensity

$$N_{fwd}(k,i) = \sum_{i' \in neigbors(i)} N_{fwd}(k+1,i')$$

$$N_{bwd}(k,i) = \sum_{i' tq\, i \in neigbors(i')} N_{bwd}(k-1,i')$$

$$N_{tot}(k,i) = N_{fwd}(k,i) \times N_{bwd}(k,i)$$

Stochastic backtracking => enumerate chains



Fragment-based ssRNA docking

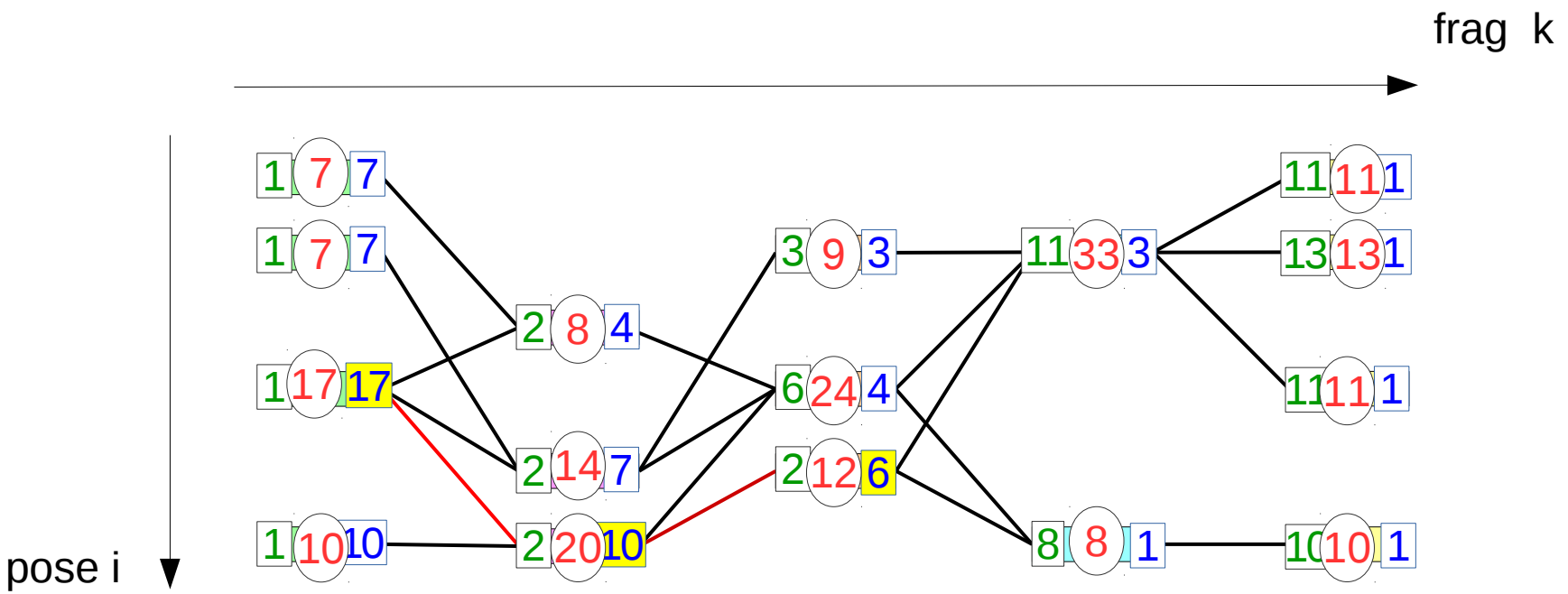
Connection propensity

$$N_{fwd}(k,i) = \sum_{i' \in neigbors(i)} N_{fwd}(k+1,i')$$

$$N_{bwd}(k,i) = \sum_{i' tq\, i \in neigbors(i')} N_{bwd}(k-1,i')$$

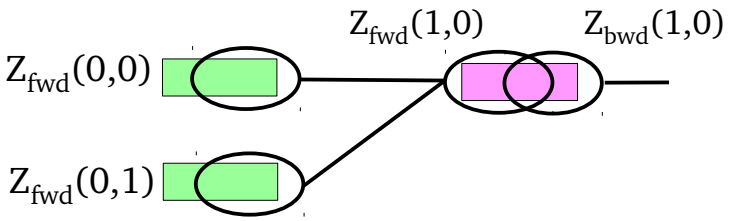
$$N_{tot}(k,i) = N_{fwd}(k,i) \times N_{bwd}(k,i)$$

Stochastic backtracking => enumerate chains



Fragment-based ssRNA docking

Weighten by Boltzmann equation

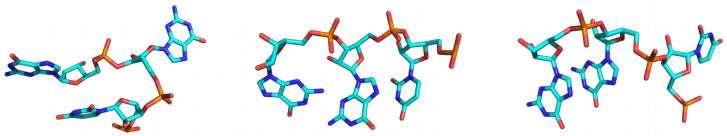


Docking scores

$$Z_{fwd}(k,i) = \sum_{j/connect(j,i)} \exp\left(\frac{E(i,j)}{RT}\right) \times Z_{fwd}(k-1,j)$$

$$P(k,i) = \frac{Z_{fwd}(k,i) \times Z_{bwd}(k,i)}{\sum_{j} P(k,j)}$$

Conformational energy



$E_{conf}(a)$ $E_{conf}(b)$ $E_{conf}(c)$

Probabilistic connectivity

connect(i, j) = 0.9 connect(i, j) = 0.5

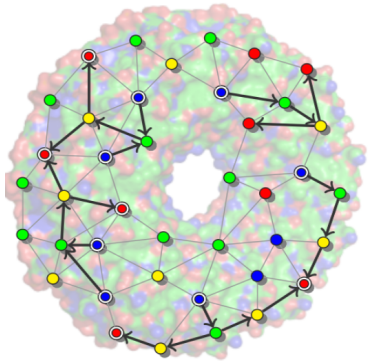
$$E(i,j) = [E_{score}(i) \times E_{conf}(i) + E_{score}(j) \times E_{conf}(j)] \times connect(i,j)$$

Avoid clashes

Color coding [Noga Alon 1995]

Self-avoiding walks in oriented graph using dynamic programming

O(nk(2e)k) complexity



collab Y. Ponty, AMIBio, LIX